# KLASIFIKACIJA

**JELENA JOVANOVIĆ**

Email: jeljov@gmail.com

Web: http://jelenajovanovic.net

# Pregled predavanja

- Šta je klasifikacija?
- Binarna i više-klasna klasifikacija
- Algoritmi klasifikacije
- Naïve Bayes (NB) algoritam
- Klasifikacija teksta primenom NB algoritma
  - Primer klasifikacije teksta korišćenjem WEKA-e
- Mere uspešnosti klasifikatora

# ŠTA JE KLASIFIKACIJA?

- Zadatak određivanja klase kojoj neka instanca pripada
  - instanca je opisana vrednošću atributa;
  - skup mogućih klasa je poznat i dat
- Klase su date kao nominalne vrednosti, npr.
  - klasifikacija email poruka: spam, not-spam
  - klasifikacija novinskih članaka: politika, sport, kultura i sl.

# BINARNA I VIŠE-KLASNA KLASIFIKACIJA

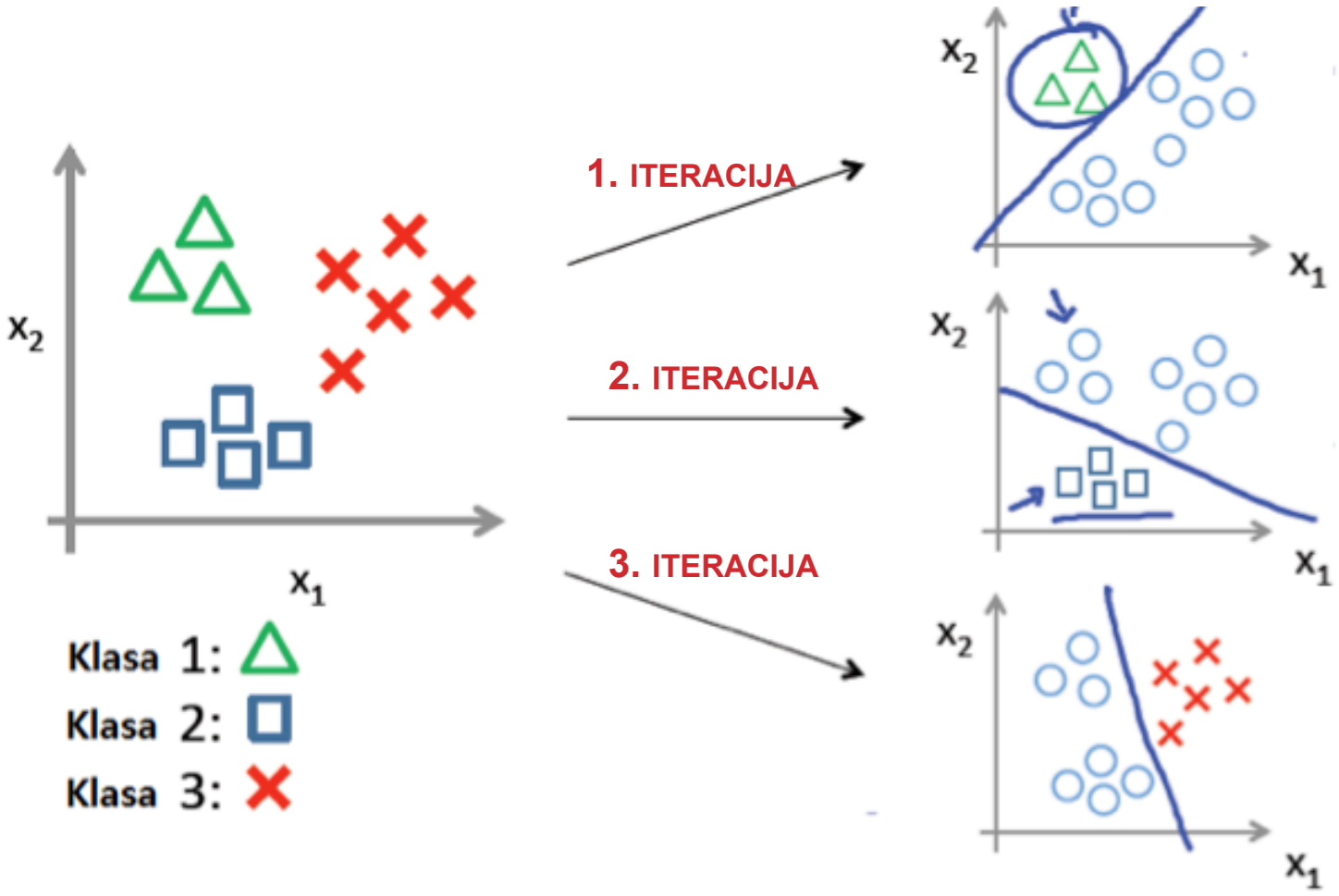
Zavisno od broja klasa, razlikujemo:

- *binarnu* klasifikaciju - postoje dve klase
- *više-klasnu* klasifikacija - postoji više klasa u koje instance treba razvrstati

Princip rada algoritma u oba slučaja je gotovo isti:

u slučaju postojanja više klasa, algoritam iterativno uči, tako da u svakoj iteraciji “nauči” da jednu od klasa razgraniči od svih ostalih

# VIŠE-KLASNA KLASIFIKACIJA




Izvor: https://www.coursera.org/course/ml

# ALGORITMI KLASIFIKACIJE

Postoje brojni pristupi/algoritmi za klasifikaciju:

- Logistička regresija
- Naïve Bayes
- Algoritmi iz grupe Stabala odlučivanja
- Algoritmi iz grupe Neuronskih mreža
- k-Nearest Neighbor (kNN)
- Support Vector Machines (SVM)
- …

# NAÏVE BAYES

# ZAŠTO BAŠ NAÏVE BAYES?

Naïve Bayes (NB) se navodi kao algoritam koji treba među prvima razmotriti pri rešavanju zadataka klasifikacije

Razlozi:

- Jednostavan je
- Ima dobre performanse
- Vrlo je skalabilan
- Može se prilagoditi za gotovo bilo koji problem klasifikacije

Occam's Razor:
"Other things being equal, simple theories are preferable to complex ones"

# PODSEĆANJE: BAYES-OVO PRAVILO

$$P(H|E) = P(E|H) * P(H) / P(E)$$

- H – hipoteza (*hypothesis*)
- E – opažaj (*evidence*) vezan za hipotezu H, tj. podaci na osnovu kojih bi trebalo da potvrdimo ili odbacimo hipotezu H
- P (H) – verovatnoća hipoteze H (*prior probability*)
- P (E) – verovatnoća opažaja tj. stanja na koje ukazuju prikupljeni podaci
- P (E | H) – (uslovna) verovatnoća opažaja E ukoliko važi hipoteza H
- P (H | E) – (uslovna) verovatnoća hipoteze H ukoliko imamo opažaj E

# BAYES-OVO PRAVILO - PRIMER

Pretpostavite sledeće:

- jednog jutra ste se probudili sa povišenom temperaturom
- prethodnog dana ste čuli da je u gradu počela da se širi virusna infekcija, ali da je verovatnoća zaraze mala, svega 2.5%
- takođe ste čuli da je u 50% slučajeva virusna infekcija praćena povišenom temperaturom
- u vašem slučaju, povišena temperatura se javlja svega par puta u godini, tako da je verovatnoća da imate povišenu temp. 6.5%

Pitanje: kolika je verovatnoća da, pošto imate povišenu temp., da imate i virusnu infekciju?

# BAYES-OVO PRAVILO - PRIMER

| Teorija | Primer |
|---|---|
| Hipoteza (H) | Imate virusnu infekciju |
| P(H) | 0.025 |
| Opažaj (evidence - E) | Imate povišenu temperaturu |
| P(E) | 0.065 |
| (uslovna) verovatnoća opažaja E ukoliko važi hipoteza H: P(E\|H) | Verovatnoća da je virusna infekcija praćena povišenom temperaturom<br>0.50 |
| (uslovna) verovatnoća hipoteze H ukoliko imamo opažaj E: P(H\|E) | Verovatnoća da pošto imate povišenu temp., da imate i virusnu infekciju<br>? |

$P(H|E) = P(E|H) * P(H) / P(E)$

$P(H|E) = 0.50 * 0.025 / 0.065 = 0.19$

# NAÏVE BAYES U KLASIFIKACIJI TEKSTA

NB je jedan od najčešće korišćenih algoritama za klasifikaciju teksta

Zadatak klasifikacije teksta: odrediti kojoj klasi (c) iz datog skupa klasa (C), dati tekst pripada

Na primer:

- tematska klasifikacija novinskih članaka
- klasifikacija tweet poruka prema iskazanom stavu (poz./neg.)

# FORMIRANJE VEKTORA ATRIBUTA

Tekst koji je predmet klasifikacije se tretira kao prost skup reči (tzv. bag-of-words)

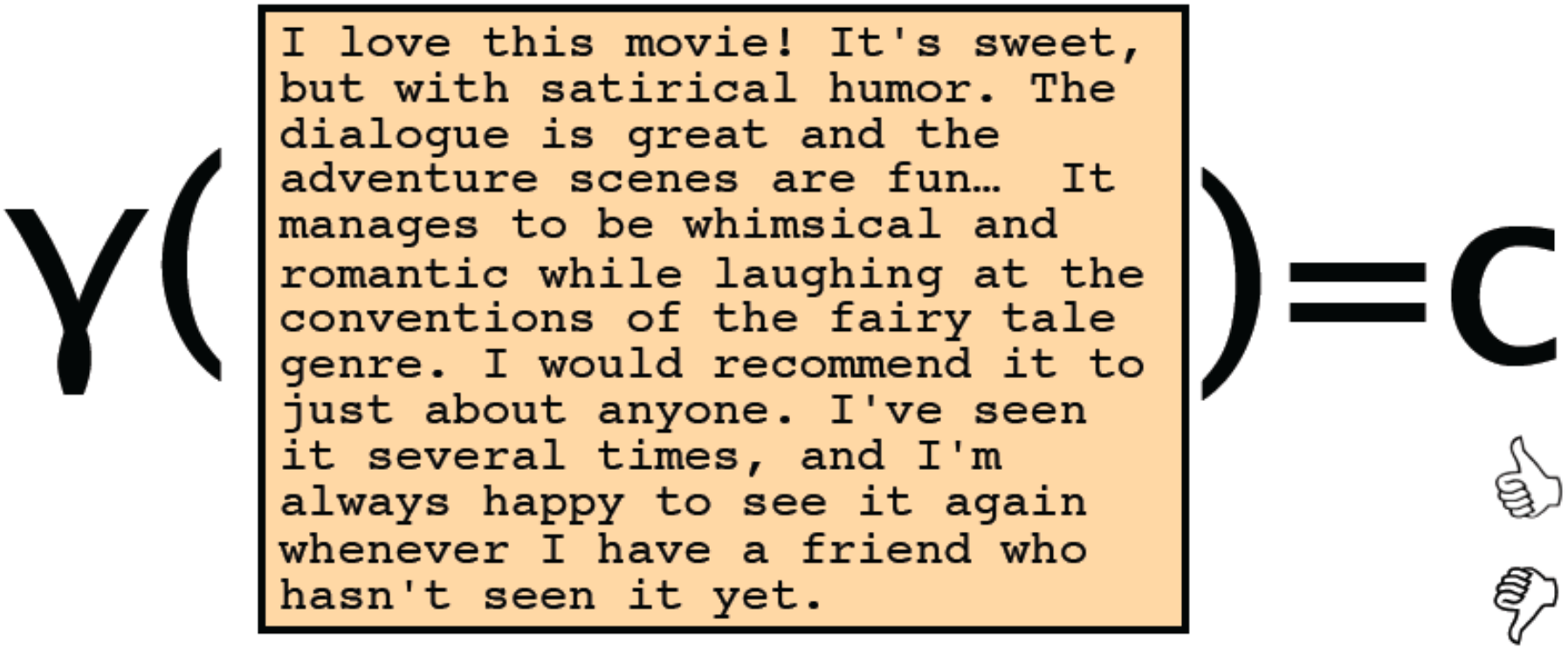
- Reči iz teksta su osnova za kreiranje atributa (features) sa kojima će NB algoritam raditi
- Postoji više načina da se reči iz teksta upotrebe za definisanje vektora atributa (feature vector) za klasifikaciju

# FORMIRANJE VEKTORA ATRIBUTA

Pristup koji se pokazao kao posebno dobar:

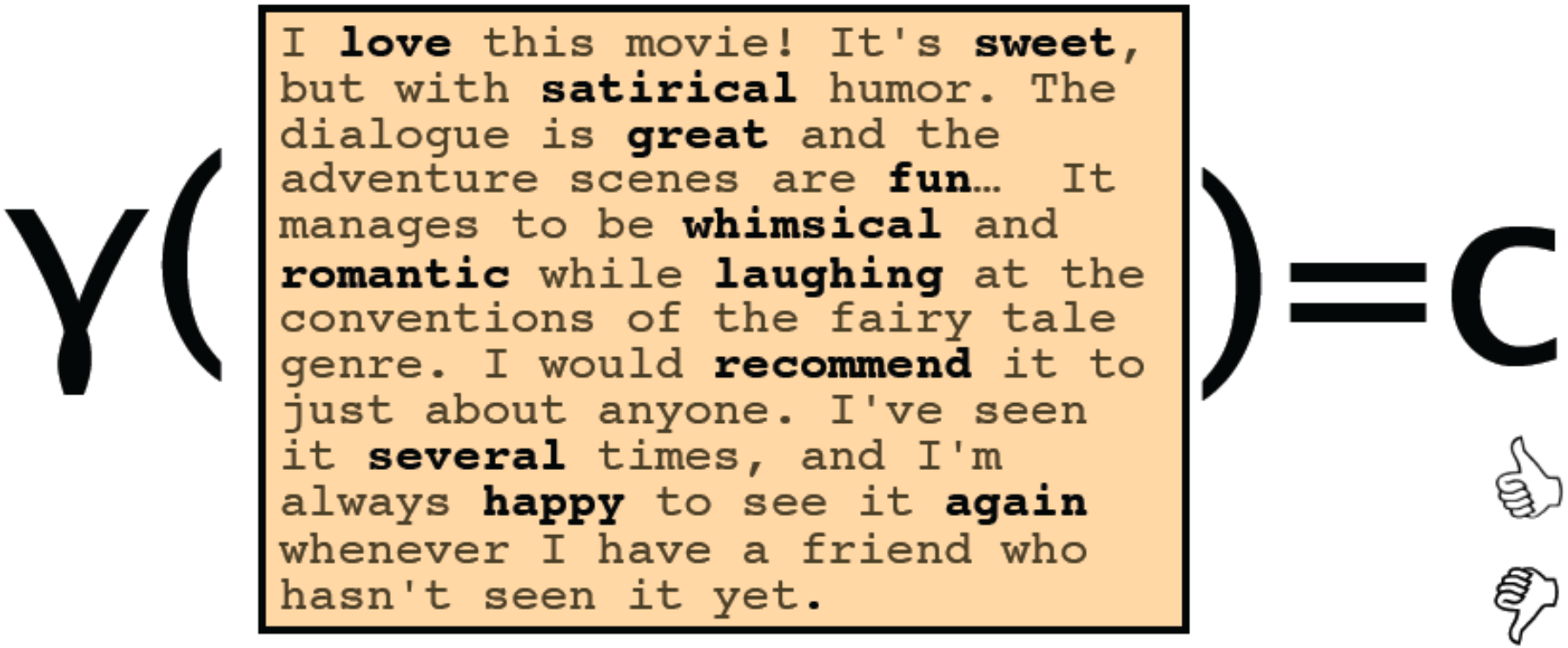
- Estrahovati reči iz dokumenata koji čine skup za trening ***D***, i formirati tzv. Rečnik ***R***;
- Za svaki dokument ***d*** iz skupa ***D*** definisati skup atributa (feature vector) na osnovu reči iz kojih se ***d*** sastoji:
  - Za svaku reč $\boldsymbol{r_i}$ iz dokumenta ***d*** uvodi se atribut $\boldsymbol{x_i}$ čija vrednost je indeks reči $\boldsymbol{r_i}$ u rečniku ***R***;
  - Atributi mogu biti kreirani za sve reči dokumenta ***d*** ili samo za one reči koje su značajne za dati zadatak klasifikacije

# FORMIRANJE VEKTORA ATRIBUTA - PRIMER




Izvor: http://www.stanford.edu/class/cs124/lec/naivebayes.pdf

# FORMIRANJE VEKTORA ATRIBUTA - PRIMER




Izvor: http://www.stanford.edu/class/cs124/lec/naivebayes.pdf

# FORMIRANJE VEKTORA ATRIBUTA - PRIMER

$\gamma($

| reči ($r_i$) | atributi ($x_i$) |
|---|---|
| $r_1$ = love | $x_1$ = 04567 |
| $r_2$ = sweet | $x_2$ = 14321 |
| $r_3$ = satirical | $x_3$ = 14007 |
| … | |
| $r_{11}$ = happy | $x_{11}$ = 02364 |
| $r_{12}$ = again | $x_{12}$ = 00012 |

$) = C$

indeks reči $r_i$ u Rečniku $R$

# NB U KLASIFIKACIJI TEKSTA

Ako je ***c*** klasa, a ***d*** dokument, verovatnoća da je upravo ***c*** klasa dokumenta ***d*** biće:

$$P(c|d) = P(d|c) * P(c) / P(d) \qquad (1)$$

Za dati skup klasa ***C*** i dokument ***d***, želimo da pronađemo onu klasu ***c*** iz skupa ***C*** koja ima najveću uslovnu verovatnoću za dokument ***d***, što daje sledeću funkciju:

$$f = \text{argmax}_{c \text{ iz } C} P(c|d) \qquad (2)$$

Primenom Bayes-ovog pravila, dobijamo:

$$f = \text{argmax}_{c \text{ iz } C} P(d|c) * P(c) \qquad (3)$$

# NB U KLASIFIKACIJI TEKSTA

$$f = \text{argmax}_{c \text{ iz } C} P(d|c) * P(c) \qquad (3)$$

Potrebno je odrediti verovatnoće P(c) i P(d|c)

P(c) se može *proceniti* relativno jednostavno: brojanjem pojavljivanja klase ***c*** u skupu dokumenata za trening ***D***

P(d|c) - verovatnoća da u klasi ***c*** zateknemo dokument ***d*** – nije tako jednostavno odrediti i tu uvodimo pretpostavke koje NB algoritam čine “naivnim”

# NB U KLASIFIKACIJI TEKSTA

Kako odrediti P(d|c)?

- dokument ***d*** predstavljamo kao skup atributa ($x_1$, $x_2$, ...,$x_n$)
- umesto P(d|c) imaćemo P($x_1$, $x_2$, $x_3$, ...$x_n$|c)
- da bi izračunali P($x_1$, $x_2$, $x_3$, ...$x_n$|c) uvodimo 2 naivne pretpostavke:
  - dokument ***d*** posmatramo kao prost skup reči (bag-of-words); tj. pozicija i redosled reči u tekstu se smatraju nevažnim
  - pojavljivanje određene reči u datoj klasi ***c*** je nezavisno od pojavljivanja neke druge reči u toj klasi

# NB U KLASIFIKACIJI TEKSTA

Uvedene pretpostavke

- dovode do značajnog gubitka informacija koje iz podataka možemo da izvučemo, ali,
- omogućuju značajno jednostavnije računanje $P(x_1, x_2,...,x_n|c)$, a time i ceo problem klasifikacije

# NB U KLASIFIKACIJI TEKSTA

Na osnovu uvedenih pretpostavki, $P(x_1, x_2,...,x_n|c)$ možemo da predstavimo kao proizvod individualnih uslovnih verovatnoća

$$P(x_1, x_2,...,x_n|c) = P(x_1|c) * P(x_2|c) * \ldots * P(x_n|c)$$

Time dolazimo do opšte jednačine NB algoritma:

$$f = \text{argmax}_{c \text{ iz } C} P(c) * \prod_{i=1,n} P(x_i|c)$$

# NB U KLASIFIKACIJI TEKSTA

Procena verovatnoća se vrši na osnovu skupa za trening, i zasniva na sledećim jednačinama:

$P(c)$ = br. dok. klase ***c*** / ukupan br. dok. u skupu za trening

$P(x_i|c)$ = br. pojavljivanja reči $r_i$ u dok. klase ***c*** / ukupan br. reči iz rečnika ***R*** u dok. klase ***c***

# Osobine NB algoritma

- Veoma brz i efikasan
- Najčešće daje dobre rezultate
  - često se pokazuje kao bolji ili bar podjednako dobar kao drugi, sofisticiraniji modeli
- Nije memorijski zahtevan
- Ima vrlo mali afinitet ka preteranom podudaranju sa podacima za trening (overfitting)
- Pogodan kada imamo malu količinu podataka za trening

# OSOBINE NB ALGORITMA

- “Otporan” na nevažne atribute
  - atributi koji su podjednako distribuirani kroz skup podataka za trening, pa nemaju veći uticaj na izbor klase
- Namenjen primarno za rad sa nominalnim atributima; u slučaju numeričkih atributa:
  - koristiti raspodelu verovatnoća atributa (tipično Normalna raspodela) za procenu verovatnoće svake od vrednosti atributa
  - uraditi diskretizaciju vrednosti atributa

# PRIMER PRIMENE NB ALGORITMA KORIŠĆENJEM WEKA FRAMEWORK-A

# Primer primene NB algoritma korišćenjem Weka framework-a

Primer je preuzet iz GitHub projekta TMWeka:

https://github.com/jmgomezh/tmweka

i raspoloživ je na sledećoj adresi:
https://github.com/jmgomezh/tmweka/tree/master/FilteredClassifier

U okviru TMWeka projekta, ima jos nekoliko interesantnih primera klasifikacije teksta primenom ML algoritama

# Mere uspešnosti klasifikatora

Neke od najčešće korišćenih metrika:

- Matrica zabune (Confusion Matrix)
- Tačnost (Accuracy)
- Preciznost (Precision) i Odziv (Recall)
- F mera (F measure)
- Površina ispod ROC krive (Area Under the Curve - AUC)

# Matrica zabune (Confusion Matrix)

Služi kao osnova za računanje mera performansi (uspešnosti) algoritama klasifikacije

| | | Predicted Class: Yes | Predicted Class: No |
|---|---|---|---|
| Actual Class | Yes | TP | FN |
| | No | FP | TN |

TP = True Positive

FP = False Positive

TN = True Negative

FN = False Negative

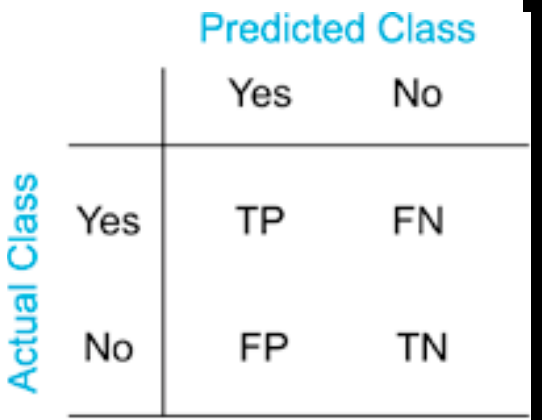
# TAČNOST (ACCURACY)

Tačnost (Accuracy) predstavlja procenat slučajeva (instanci) koji su uspešno (korektno) klasifikovani

**Accuracy** = (TP + TN) / N

gde je:

- TP – True Positive; TN – True Negative
- N – ukupan broj uzoraka (instanci) u skupu podataka



# Tačnost (Accuracy)

U slučaju vrlo neravnomerne raspodele instanci između klasa (tzv. skewed classes), ova mera je nepouzdana

Npr. u slučaju klasifikacije poruka na spam vs. not-spam, možemo imati skup za trening sa 0.5% spam poruka

Ako primenimo “klasifikator” koji svaku poruku svrstava u not-spam klasu, dobijamo tačnost od 99.5%

Očigledno je da ova metrika nije pouzdana i da su u slučaju skewed classes potrebne druge metrike

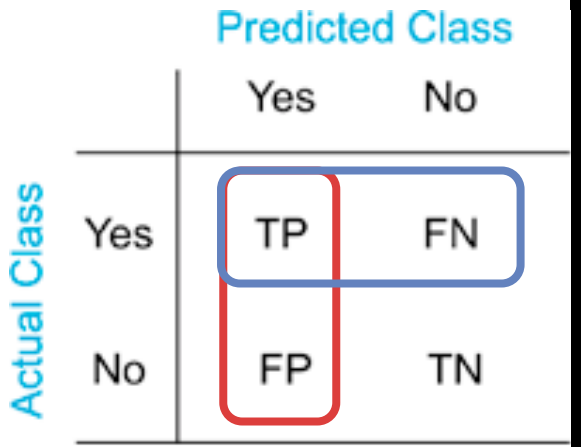
# Preciznost (Precision) i Odziv (Recall)

**Precision** = TP / no. predicted positive = TP / (TP + FP)

Npr. od svih poruka koje su *označene kao spam* poruke, koji procenat čine poruke koje su stvarno spam

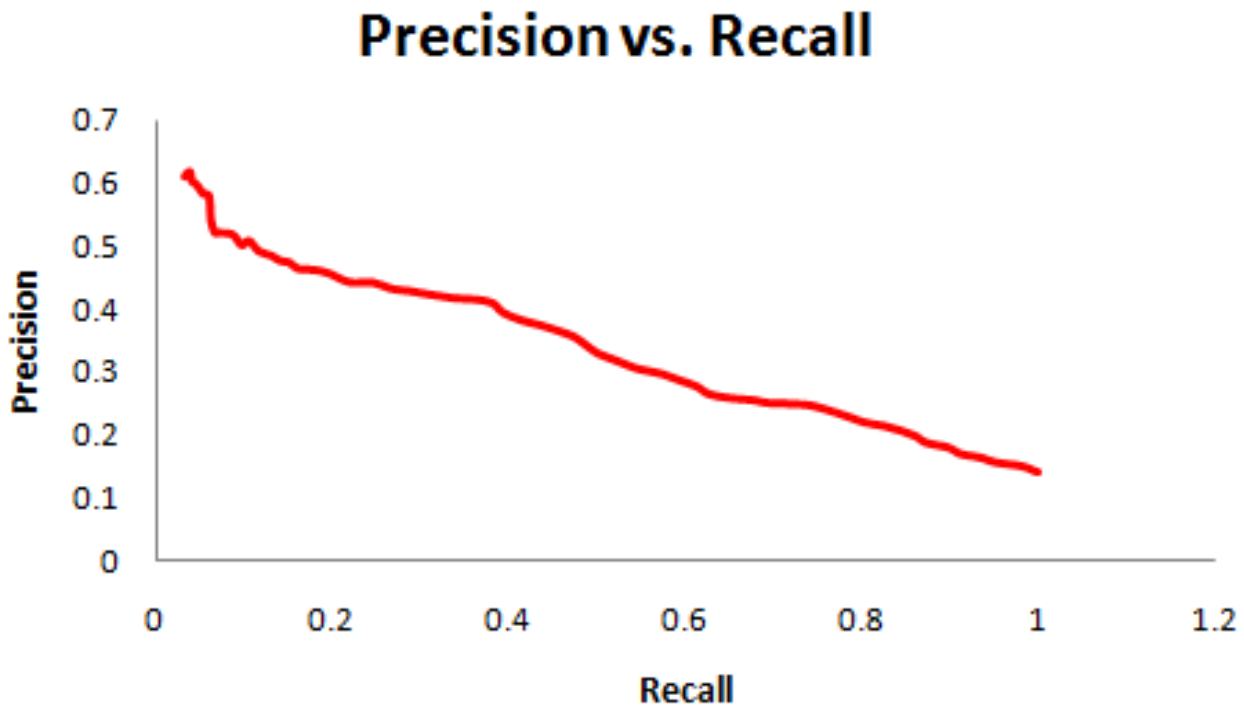
**Recall** = TP / no. actual positive = TP/ (TP + FN)

Npr. od svih poruka koje su *stvarno spam* poruke, koji procenat poruka je detektovan/klasifikovan kao spam



# PRECIZNOST I ODZIV

U praksi je nužno praviti kompromis između ove dve mere: ako želimo da povećamo Odziv, smanjićemo Preciznost, i obrnuto.




Izvor:
http://groups.csail.mit.edu/cb/struct2net/webserver/images/prec-v-recall-v2.png

# F MERA (F MEASURE)

F mera kombinuje Preciznost i Odziv i omogućuje jednostavnije poređenje dva ili više algoritama

$$F = (1 + \beta^2) * Precision * Recall / (\beta^2 * Precision + Recall)$$

Parametar $\beta$ kontroliše koliko više značaja će se pridavati Odzivu u odnosu na Preciznost

U praksi se najčešće koristi tzv. F1 mera („balansirana“ F mera) koja daje podjednak značaj i Preciznosti i Odzivu:

$$F1 = 2 * Precision * Recall / (Precision + Recall)$$

# POVRŠINA ISPOD ROC KRIVE

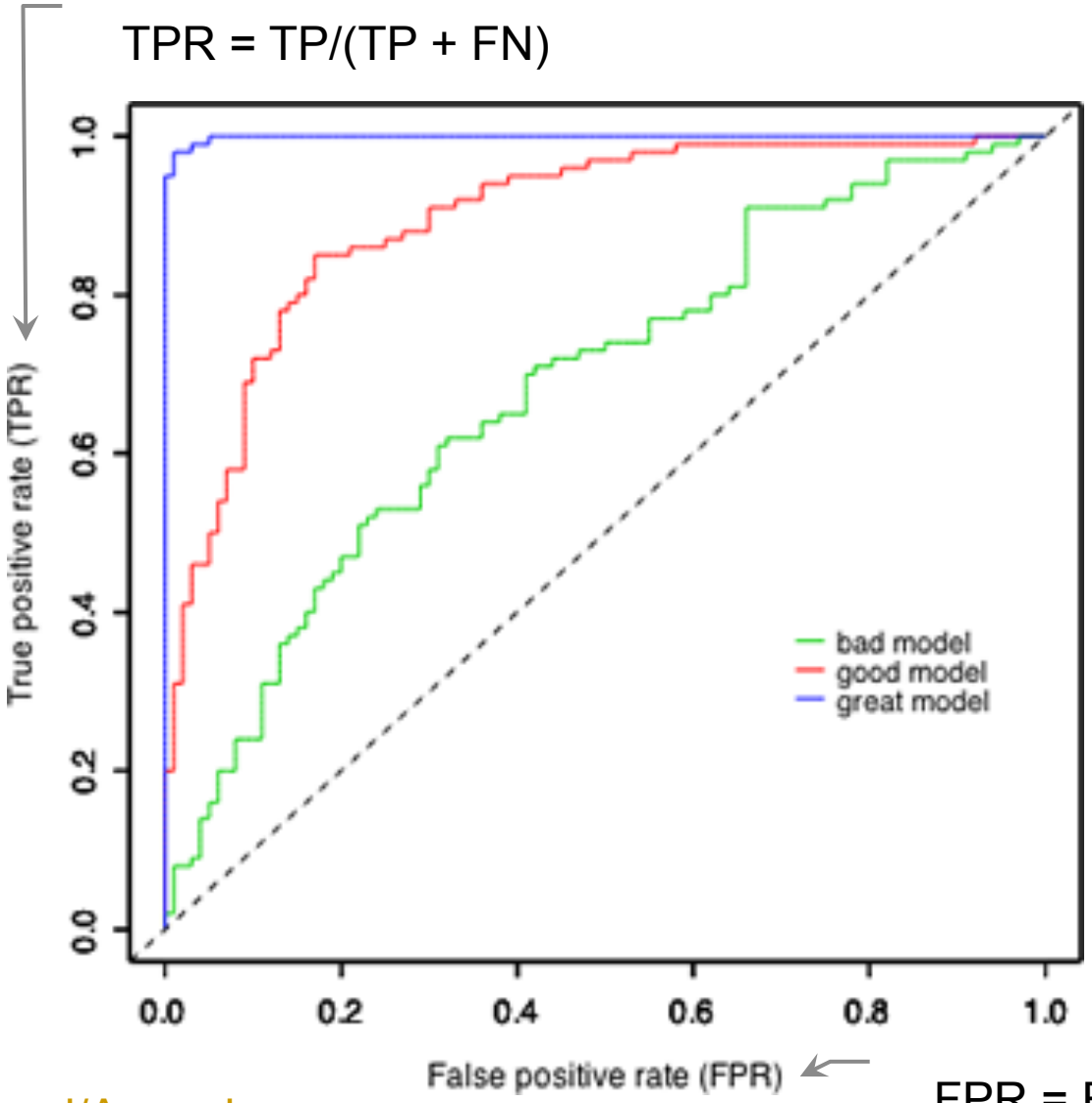
Površina ispod ROC* krive – Area Under the Curve (AUC):

- meri diskriminacionu moć klasfikatora tj. sposobnost da razlikuje instance koje pripadaju različitim klasama
- primenjuje se za merenje performansi binarnih klasifikatora
- vrednost za AUC se kreće u intervalu 0-1
- za metodu slučajnog izbora važi da je AUC = 0.5; što je AUC vrednost klasifikatora > 0.5, to je klasifikator bolji
  - 0.7–0.8 se smatra prihvatljivim; 0.8–0.9 jako dobrim; sve > 0.9 je odlično

*ROC = Receiver Operating Characteristic;
http://en.wikipedia.org/wiki/Receiver_operating_characteristic

# POVRŠINA ISPOD ROC KRIVE

TPR = TP/(TP + FN)




FPR = FP/(FP + TN)

Izvor: http://goo.gl/Aeauuh

# ZAHVALNICE I PREPORUKE

# PREPORUKE I ZAHVALNICE

## MACHINE LEARNING @ STANFORD

- Coursera: https://www.coursera.org/course/ml
- Stanford YouTube channel: http://www.youtube.com/view_play_list?p=A89DCFA6ADACE599

## MACHINE LEARNING @ Carnegie Melon University

- Predavanja Andrew W. Moore-a, posebno, lekcija na temu Bayes-ovog pravila i Naïve Bayes algoritma: http://www.autonlab.org/tutorials/prob_and_naive_bayes.pdf

# Preporuka

## Data Stories podcasts

- http://datastori.es/
- posebno podcast #27 na temu “Big Data Skepticism”
- studija pomenuta u podcast-u #27 na temu predikcije karakteristika korisnika FB-a, samo na osnovu njihovih FB Likes:
  http://www.pnas.org/content/early/2013/03/06/1218772110.full.pdf

(Anonimni) upitnik za vaše kritike, komentare, predloge:

http://goo.gl/cqdp3l